УДК 004.934.1’1

В.Ю. Шелепов, А.В. Ниценко
Институт проблем искусственного интеллекта
Мин образования и науки и НАН Украины
shel@iai.donetsk.ua

Структурная классификация слов русского языка. Новые алгоритмы сегментации речевого сигнала и распознавания некоторых классов фонем

Компьютерное распознавание устной речи – традиционная задача искусственного интеллекта. Ею стали с энтузиазмом заниматься еще на заре возникновения ИНФОРМАТИКИ как науки так же, как и задачей автоматического перевода с одного языка на другой. Однако по прошествии нескольких десятилетий результаты многочисленных исследовательских групп и в той и в другой области остаются весьма скромными. Естественные языки оказались значительно более сложным объектом, чем это казалось с самого начала. В частности, распознавание устной речи человеком в значительной степени основывается на умственной деятельности, связанной с осмыслением в реальном времени содержания произносимого. Известно, что без этого человек способен правильно идентифицировать не более 30 % услышанного потока фонем. Однако он способен повторить слово, которое четко произнесено на незнакомом языке (конечно, интерпретируя звуки в рамках привычной для него фонетической системы). И это означает, что пофонемное распознавание все-таки должно приниматься за основу, если мы не хотим ограничиться распознаванием слов по эталонам, то есть работать на уровне звуковых иероглифов.

Мы будем использовать термин «пофонемное распознавание», хотя на самом деле при распознавании однозначно определяются не фонемы. Насколько нам известно, единого понимания термина «фонема» не существует. Если придерживаться точки зрения московской школы, - это фонетическая единица, связанная не только со звучащей речью, но и с морфологией. С этой точки зрения в словах МОРОЗЫ и МОРОЗ присутствует одна и та же фонема «З», представленная разными звуками, ввиду того, что в силу позиционного чередования, «З» на конце слова оглушается и реализуется звуком «С». Не очень уместно говорить и о распознавании аллофонов, так эта более тонкая градация: например, «С» в слогах «СА» и «СО» представлена разными аллофонами, но в задачах «Речь-Текст» их естественно желательно отождествлять. Возможно, таким образом, объекты, которые должны распознаваться, наиболее уместно обозначать не очень строгим, но емким русским термином «Звуки речи». Однако, мы будем использовать традиционную терминологию и говорить о «пофонемном распознавании».

Упрощенная транскрипция русских слов (автоматическое построение)

Очевидно, что пофонемное распознавание слов, список которых задан в текстовом виде, требует предварительного создания транскрипции каждого из этих слов. Эта процедура, разумеется, должна осуществляться автоматически. В отличие от синтеза речи, когда транскрипция должна быть максимально подробной, при распознавании допустима и даже желательна упрощенная транскрипция. Мы используем простой транскриптор, который позволяет по мере накопления опыта модифицировать систему транскрипции путем простейших изменений в управляющем файле с целью учета ранее не учтенных транскрипционных ситуаций. В качестве транскрипционных знаков использованы в основном соответствующие русские буквы. Исключение составляют символ j, обозначающий согласный типа того, что возникает первым при произнесении названий букв «е», «ё», «ю», «я», а также символы l, t, h, -, обозначающие соответственно Л мягкое, Т мягкое, Х мягкое и результат соседства двух глухих взрывных. Транскриптор реализован как программа, заменяющая одни символы другими в соответствии с правилами, содержащимися в упомянутом управляющем файле. Вот его содержание на сегодняшний день:

ль=l, лj=lj, ле=le, ли=lи, лю=lю, ля=lя, л\е=l\е, л\ё=l\ё, л\и=l\и, л\ю=l\ю, л\я=l\я, ть=t, тj=tj, те=te, ти=ти, тю=тю, тя=тя, т\е=t\е, т\ё=t\ё, т\и=t\и, т\ю=t\ю, т\я=t\я, хь=h, хj=hj, хе=he, хи=hи, хю=hю, хя=hя, х\е=h\е, х\ё=h\ё, х\и=h\и, х\ю=h\ю, х\я=h\я,

ъе=jе, ъё=jё, ъю=jю, ъя=jя, ье=jе, ьё=jё, ью=jю, ья=jя, ъ\е=j\е, ъ\ё=j\ё, ъ\ю=j\ю, ъ\я=j\я, ь\е=j\е, ь\ё=j\ё, ь\ю=j\ю, ь\я=j\я,

а\е=аj\е, а\ё=аj\ё, а\ю=аj\ю, а\я=аj\я, е\е=еj\е, е\ё=еj\ё, е\ю=еj\ю, е\я=еj\я, и\е=иj\е, и\ё=иj\ё, и\ю=иj\ю, и\я=иj\я, о\е=оj\е, о\ё=оj\ё, о\ю=оj\ю, о\я=оj\я, у\е=уj\е, у\ё=уj\ё, у\ю=уj\ю, у\я=уj\я, ы\е=ыj\е, ы\ё=ыj\ё, ы\ю=ыj\ю, ы\я=ыj\я, ю\е=ю\jе, ю\ё=юj\ё, ю\ю=юj\ю, ю\я=юj\я, я\е=яj\е, я\ё=яj\ё, я\ю=яj\ю, я\я=яj\я, \ее=\еjе, \ею=\еjю, \ея=\еjя, \ие=\иjе, \ию=\иjю, \ия=\иjя, \юе=\юjе, \юё=\юjё, \юю=\юjю, \юя=\юjя, \яе=\яjе, \яю=\яjю, \яя=\яjя,

#е=jе, #ю=jю, #я=jя, #\е=j\е, #\ё=j\ё, #\ю=j\ю, #\я=j\я,

\а=1, а=q, 1=\а, \о=2, о=q, 2=\о, \э=3, э=и, 3=\э, \е=4, е=и, 4=\е, \я=5, я=и, 5=\я,

б#=п, в#=ф, г#=к, д#=т, ж#=ш, з#=с, бь#=п, вь#=ф, дь#=t, жь#=ш, зь#=с,

кп=-, кт=-, кф=-, кх=-, пк=-, пт=-, пф=-, пх=-, тк=-, тп=-, тф=-, тх=-, фк=-, фп=-, фт=-, фх=-, хк=-, хп=-, хт=-, хф=-,

бк=-, бп=-, бс=пс, бт=-, бф=-, бх=-, бц=пц, бч=пч, бш=пш, бщ=пщ, вк=-, вп=-, вс=фс, вт=-, вф=-, вх=-, вц=фц, вч=фч, вш=фш вщ=фщ гк=-, гп=-, гс=кс, гт=-, гф=-, гц=-, гч=-, гх=-, гш=кш, гщ=кщ, дк=-, дп=-, дс=тс, дт=-, дф=-, дх=-, дц=тц, дч=тч, дш=тш, дщ=тщ, жк=-, жп=-, жс=пс, жт=-, жф=-, жх=-, жц=пц, жч=пч, жш=пш, жщ=пщ, зк=-, зп=-, зс=пс, зт=-, зф=-, зх=-, зц=пц, зч=пч, зш=пш, зщ=пщ,

#к=, #п=, #т=, #-=,

кб=гб, кг=г, кд=гд, кж=гж, кз=гз, пб=б, пг=бг, пд=бд, пж=бж, пз=бз, тб=дб, тьб=дб, тг=дг, тьг=дг, тд=д, тьд=д, тж=дж, тз=дз, сб=зб, сьб=зьб, сг=зг, сьг=зьг, сд=зд, сьд=зьд,

здн=зн, дц=ц тц=ц, жч=щ, зж=ж, сш=ш, сч=щ,

же=жэ, жи=жы, жю=жу, жя=жы, ж\е=ж\э, ж\ё=ж\о, ж\и=ж\ы, ж\ю=ж\у, ж\я=ж\я, ше=шэ, ши=шы, шю=шу, шя=шы, ш\е=ш\э, ш\ё=ш\о, ш\и=ш\ы, ш\ю=ш\у, ш\я=ш\а, це=цэ, ци=цы, цю=цу, ця=ца, ц\е=ц\э, ц\ё=ц\о, ц\и=ц\ы, ц\ю=ц\у, ц\я=ц\а,

ндс=нс, нтс=нс, стн=сн, стс=с, стц=сц, здн=зн, здц=сц, рдц=рц, рдч=рч,

й=, ие=и,

ь=,

Поясним приведенный перечень правил. Каждое правило записано в виде двух частей, соединенных знаком =. Слева стоят исходные символы буквенной записи слова, справа – символы которыми они заменяются в транскрипции. Значок \ означает ударение.

Первая группа служит для выделения мягкого «л», мягкого «т» и мягкого «х» в то время, как мягкие варианты всех остальных согласных отождествляются в нашей транскрипции с соответствующими твердыми. Причина особого подхода к «л» – принципиальное отличие артикуляции твердого и мягкого. «л». Что касается «т» мягкого, - оно, в отличие от «т» твердого, по своей структуре близко к аффрикате. Наконец, мягкое «х», в отличие от «х» твердого, является не «паузообразной», а «шипящеобразной» фонемой.

Вторая группа правил отражает фонетическую роль мягкого и твердого знака перед Е, Ё, Ю, Я. Их наличие приводит к появлению при произношении согласного j.

Третья группа отражает произношение сочетаний гласных с гласными Е, Ё, Ю, Я.

Четвертаяя группа описывает произношение Е, Ё, Ю, Я в начале слова (# -знак абзаца и, следовательно, знак начала слова, если слова в текстовом файле расположены в столбец).

Пятая группа отражает произношение гласных А, О, Э, Е, Я, когда они стоят в безударной позиции. Поскольку транскриптор работает по принципу замены, приходится предварительно переименовывать ударные гласные, а затем возвращать им прежние обозначения.

Шестая группа – оглушение звонкой согласной в конце слова.

Седьмая группа – принцип транскрибирования двух идущих подряд глухих взрывных и подобных ситуаций.

Восьмая группа – оглушение звонкой согласной перед глухой взрывной, шипящей и аффрикатой.

Появление девятой группы вызвано неспособностью нашего распознавателя обнаруживать на сегодняшний день глухие взрывные в начале слова.

Десятая – озвончение глухих согласных перед звонкими согласными.

Одиннадцатая – отражение произносительной нормы в словах типа «тридцать», «мужчина», «позже», «сшить», «счет».

Двенадцатая группа отражает влияние твердой согласной на последующую гласную.

Тринадцатая группа отражает фонетическую норму, связанную с непроизносимыми согласными.

Четырнадцатая группа связана с отсутствием у нас возможности обнаруживать глухие согласные в конце слова.

Пятнадцатая группа упрощает транскрипцию. По нашему опыту эти упрощения пока не мешают, а помогают распознаванию.

Исключение из транскриптора на заключительном этапе мягкого знака соответствует идеологии отождествления твердых и мягких согласных. В случае, когда за ними следует гласная, мы распознаем ситуацию, ориентируясь на нее, и в результате опосредованно получаем информацию о твердости или мягкости согласной. В конеце слова нужно выяснять твердость или мягкость согласной непосредственно. Наше соглашение об отождествлении твердых и мягких согласных не позволяет этого и это существенный недостаток предлагаемого транскриптора.В оправдание можно лишь сказать, что он отражает наши возможности в области распознавания на сегодняшний день. Отметим также, что в программу транскриптора заложена автоматическая замена удвоенных символов одинарными.

Структурная классификация слов русского языка и пофонемное распознавание русской речи

Первоначальные результаты этого раздела получены при участии Е.Е. Федорова.
Разобьем все участвующие в транскрипциях фонемы на несколько естественных классов:

1) аеёоэя
2) иуыю
3) бвгд
4) jлlмн
5) жз (1)
6) р
7) пкт-t
8) сшщцчh
9) фх

Первый – компактные гласные, второй – диффузные гласные, третий – взрывные голосовые согласные плюс «в», четвертый – основные сонорные согласные плюс j, пятый – голосовые согласные с шумом, шестой – «р», седьмой - то, что при произношении выступает как пауза в слове, восьмой - шипящие, аффрикаты и мягкое «х», девятый – глухие протяженные «ф» и «х».

Пусть далее есть большой словарь, который размечаем, сопоставив каждому слову его транскрипцию и заменив каждый транскрипционный символ номером его класса. При этом, если цифра повторяется подряд несколько раз, заменяем ее одной. Вот отрезок размеченного таким образом словаря Зализняка:

машина 418241
машинальность 418241418
машинальный 41824142
машинизация 4182425182
машинизировать 418242526131
машинизироваться 418242526131782
машинист 4182428
машинистка 418242871

Про слова с одинаковой разметкой будем говорить, что они имеют одинаковую структуру. Таким образом, структура – это некая модель чередования гласных, согласных, шипящих и т. д. Оказывается, что в русском языке слов с одинаковой структурой относительно мало. Вот, например, все слова со структурой 314251:

долежать 314251
долезать 314251 (2)
долизать 314251
донизать 314251

А вот все слова со структурой 6241871:

крылечко 34353
приласкать 34353
примазка 34353
примочка 34353
риноскоп 34353
ромашка 34353
рюмочка 34353

И так далее. На почти 100-тысячный словарь Зализняка максимальное число слов с одинаковой структурой (171) равно106, то есть около 0,1 процента. Причем это фактически исключительный случай. Все остальные структуры содержат единицы, или несколько десятков слов. Это доказано нами с помощью программы, которая автоматически делает разметку и выбор слов с одинаковой структурой. Причем выбор классов (1) можно менять.

Для пофонемного распознавания из этого вытекает следующее. Будем при произнесении слова выводить в качестве сокращенного списка кандидатов на распознавание только те слова, которые имеют ту же структуру, что и распознанное. Например, для слова «долежать» это будет список (2). Теперь допустим, что машина ошибочно распознала 1-ую фонему и вывела слово «волежать». Это повлечет за собой тот же список кандидатов (2). А теперь заменим каждую фонему произвольным набором фонем того же класса и получим что-то вроде

«дгвбоаэмлнеяёжзжзаоять»

От слова «долежать» здесь не осталось фактически ничего. Тем не менее, этот результат распознавания породит все тот же список (2) всего лишь из 4-х кандидатов. Таким образом, верное распознавание последовательности классов при любых ошибках внутри классов приводит к сокращению числа кандидатов на распознавание в тысячи раз.

Организация записи речевого сигнала, определение его начала и конца

За основу принимается 8-битная оцифровка звукового сигнала с частотой дискретизации 22050 Гц, так что его значения имеют $2^8 = 256$ градаций: от 0 до 255.

Предполагается использование системы в лабораторных условиях, при отсутствии существенного внешнего шума. При настройке системы записи по нажатии

соответствующей кнопки записывается 30000 отсчетов «тишины» и в записанном сигнале анализируются последовательные отрезки по 256 отсчетов в каждом. Для каждого из них вычисляется отношение

$$V / C, \tag{3}$$

где

$$V = \sum_{i=1}^{256} \left| x_{i+1} - x_i \right| \tag{4}$$

- численный аналог полной вариации, C - число точек постоянства, то есть моментов времени, для которых в следующий момент величина сигнала остается неизменной. Автоматически определяется значение величины (3), характерное для используемой звуковой карты, как наиболее часто встречающееся в массиве значений. Оно увеличивается на 0,1 и заносится в управляющий файл $recorder.ini$ под именем $StartPorog$, а результат, увеличенный в 10 раз, – под именем $EndPorog$.

При распознавании речи по нажатии кнопки записи компьютер начинает записывать сигнал, поступающий с микрофона и вычислять для последовательных отрезков по 256 отсчетов величину (3), Определяется момент, после которого эта величина впервые не менее пяти раз подряд превышает $StartPorog$ и, начиная с него, в буфер1 заносится 30000 отсчетов, после чего запись останавливается. Далее выполняется аналогичная операция в обратном направлении от конца к началу, определяется момент, когда величина (3) не менее пяти раз подряд превышает $EndPorog$ и проставляется метка конца речи. Отрезок от начала буфера1 до метки заносится в буфер 2. Поскольку превышение $StartPorog$ может произойти не за счет появления речи, а за счет случайного шума, содержимое буфера2 анализируется на наличие речи путем вычисления последовательности квазипериодов. Если найдется не менее 5 идущих подряд квазипериодов, значения которых превышают заданную пороговую величину, то содержимое буфера 2 считается речью и передается в окончательный буфер3, как объект для визуализации и дальнейшего распознавания. Упомянутый порог определяется высотой голоса диктора, для тенора его можно взять (при используемой частоте дискретизации) равным 100.

Новые алгоритмы сегментации речевого сигнала

Проблема, упомянутая в заглавии настоящего раздела является составной частью задачи пофонемного распознавания отдельно произносимых слов и слитной речи. При графическом отображении сигнала, значение 128 соответствует так называемой средней линии. По вопросам сегментации авторами опубликованы работы [1-4]. Описанные в них методы основаны на фильтрации сигнала и естественной процедуре разбиения его на высокоамплитудные участки, соответствующие гласным, и низкоамплитудные участки, соответствующие согласным звукам. Основным орудием при этом была величина среднего отклонения от средней линии на окне в 256 отсчетов.

$$E = \sqrt{\frac{\sum_{i=1}^{256} |x_i - 128|^2}{256}} \qquad (5)$$

Описанная в [4] процедура сегментации хорошо справляется с разделением между собой соседних фонем, одна из которых является согласной, а другая - одной из компактных гласных *А*,*О*,*Э*. Однако при этом возникают трудности с разделением соседних *М*, *И*, что нередко приводит к ошибкам сегментации. Для того чтобы избавиться от этих ошибок, предлагается вместо величины (5) использовать численный аналог полной вариации речевого отрезка как функции времени (4), также вычисляемый для окна размером в 256 отсчетов. Эта величина различает следующие друг за другом гласные и согласные фонемы так же хорошо, как и величина (5), и значительно лучше работает с упомянутыми М, И. Отметим, что при замене величины (5) величиной (4) роль фильтрации сигнала выполняет операция его трехточечного сглаживания

$$y_i = \frac{y_{i-1} + y_i + y_{i+1}}{3}, \qquad (6)$$

повторенная при необходимости нужное число раз.

Далее, ранее мы находили метки между участками фонем одну за другой, используя интервал поиска, начинающийся на месте только что найденной очередной метки. Вторая трудность состояла в отсутствии соображений относительно длины каждого такого интервала. Для преодоления этой трудности предлагается использовать скользящий интервал длиной 256×20 отсчетов (20 окон по 256 отсчетов в каждом). Итак, процедура сегментации приобретает следующий вид. Сигнал разбивается на отрезки по 256 отсчетов и на каждом из них вычисляется величина (4). Далее от начала сигнала последовательно берется 20 таких отрезков и вычисляется среднее значение соответствующих величин - порог. Тем отрезкам, для которых величина больше среднего присваивается символ «В» (выше порога), остальным символ «Н» (ниже порога). Чтобы устранить случайные единичные включения для каждого i-го элемента полученной символьной последовательности S выполняется две обработки:

1-я обработка:

если $s[i\text{-}1] = s[i+1]$ и $s[i] \neq s[i-1]$, то полагается

$$s[i] = s[i-1]. \qquad (7)$$

2-я обработка:

если $s[i] = s[i+3]$ и $s[i+1] \neq s[i]$, $s[i+2] \neq s[i]$, то полагается

$$s[i+1] = s[i] \quad \text{и} \quad s[i+2] = s[i]. \tag{8}$$

Затем интервал в 20 окон, на котором выполняются описанная процедура, сдвигается вправо на одно окно и процедура повторяется. Это происходит до тех пор, пока упомянутый интервал находится в пределах сигнала. В результате возникает таблица следующего вида (В - значение выше среднего, Н – значение ниже среднего на интервале)

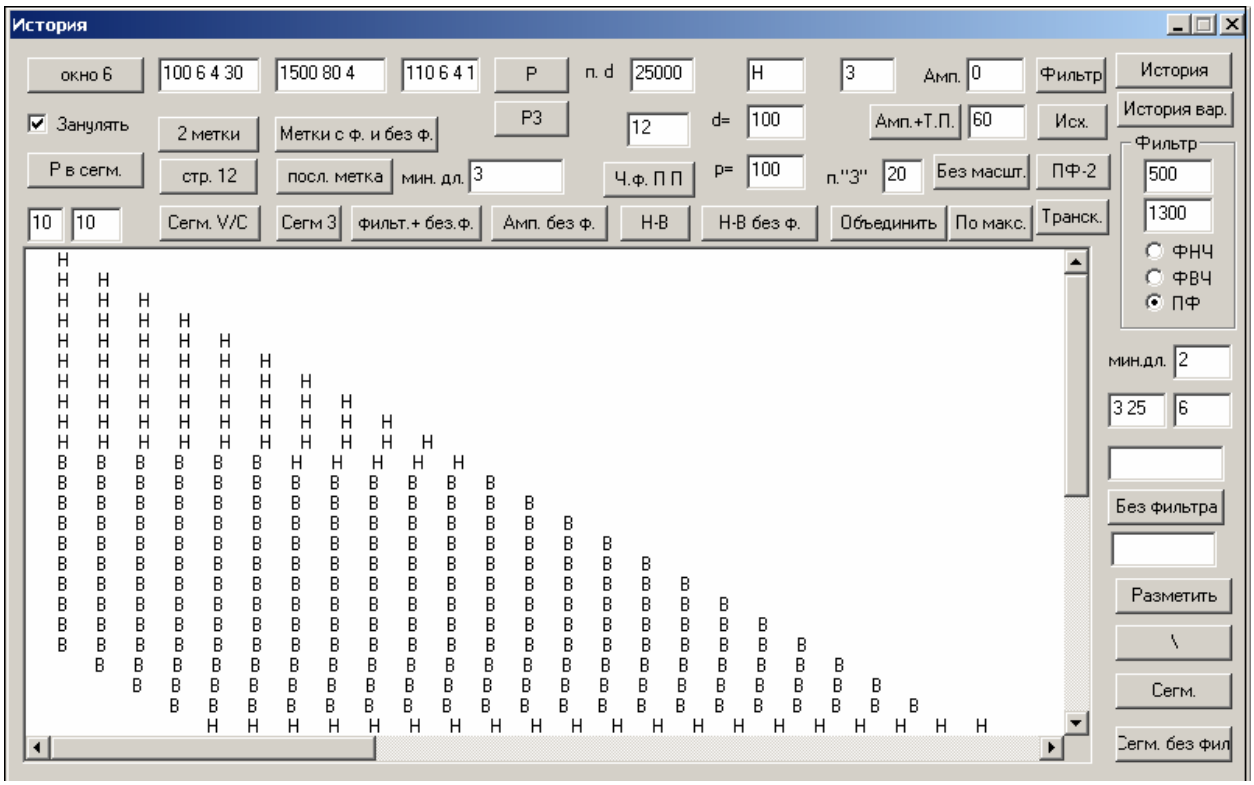

Рисунок 1 – Таблица, используемая при сегментации

Предположим, что первый столбец, как в приведенном примере таблицы, начинается с «Н». Если далее в этом столбце встречается последовательность «В», а затем снова последовательность «Н», то в начале окна, которому соответствует первое «В», ставим метку, отделяющую участок первой фонемы от участка второй. Если указанного чередования «Н» и «В» - массивов нет, анализируем следующий столбец, и так далее до тех пор, пока на находим столбец, где такое чередование найдется. Тем самым мы определяем первый «Н-В» переход. Затем, двигаясь слева направо, находим первый столбец, который начинается с «В» и таким же образом ищем «В-Н» переход и так далее. Если из-за приближения к концу сигнала искомого чередования трех массивов не обнаруживается, то ограничиваемся двумя и на их границе ставим метку, которую считаем разделяющей последние две фонемы слова. Случай, когда первый столбец начинается с «В» исчерпывается аналогично.

Вот результат для слова «мимо», отсегментированного в соответствии с только что описанными алгоритмами:

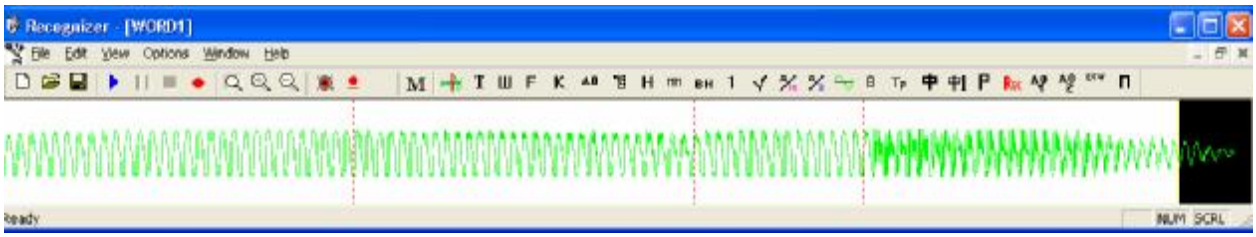

Рисунок 2 – Сегментация слова “мимо”

Трудность, возникающая при использовании величины (4), состоит в том, что для шипящих *С,Ш,Щ,* а также для фонем *Ж,З* величина (4) является относительно большой, существенно превосходящей ее значения для чисто голосовых согласных. Поэтому участки, соответствующие указанным фонемам приходится определять заранее с помощью специальных алгоритмов, их описанию посвящены следующие два пункта настоящего отчета. Когда это сделано, величины (4), отвечающие соответствующим окнам, полагаются равными нулю и после этого участки фонем *С,Ш,Щ,З,Ж* оказываются при работе с таблицей вида 1 в числе включенных в «Н» - участки. Если при этом возникают метки, отстоящие от уже существующих меток для *С,Ш,Щ,Ж,З,* на расстояние не более, чем в три окна, они рассматриваются как лишние и удаляются. Описанные в следующих пунктах алгоритмы выделяют также участки пауз, отвечающие звукам *К,П,Т*, а также звукам *Х,Ф,Ц,Ч. h,t.* С ними мы при работе с таблицей вида 1 поступаем так же, как с *С,Ш,Щ,З,Ж:* полагаем величины (4), отвечающие соответствующим окнам, равными нулю.

Выделение глухих согласных

В данном пункте предлагается новый алгоритм выделения согласных С,Ш,Щ,Ц,Ч,Ф,Х,h,П,К,Т,t произнесение которых происходит без участия голосовых связок. В основе его лежит обработка сигнала полосовым фильтром с интервалом пропускания от 100 до 200 Гц. Вот как выглядит запись слова «ОСА» до и после такой фильтрации. Отфильтрованный сигнал пронормирован так, чтобы его максимальное значение равнялось 256 (либо минимальное значение равнялось нулю):

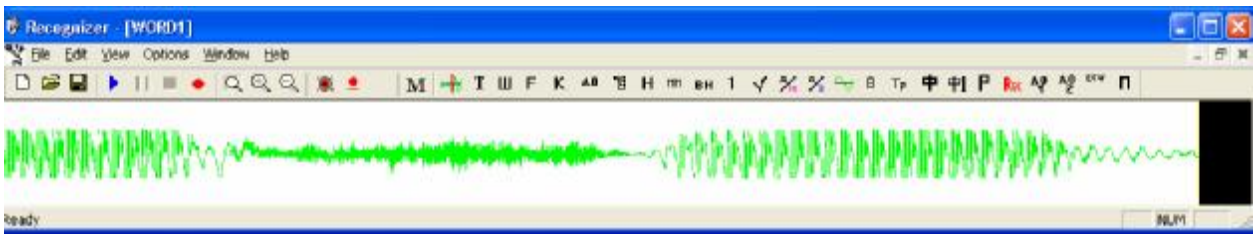

Рисунок 3 – Визуализация слова “оса”

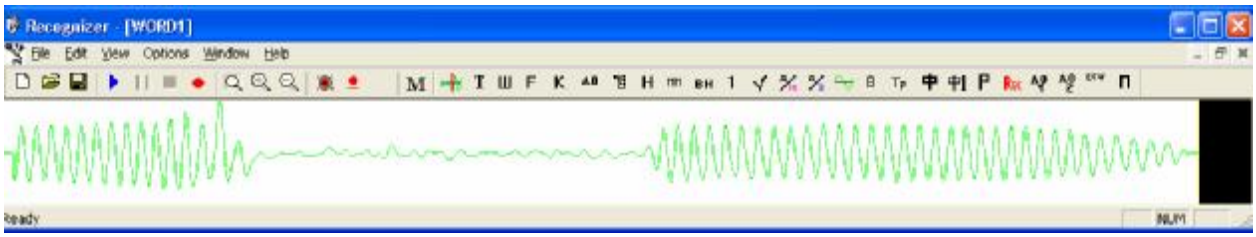

Рисунок 4 – Визуализация того же слова после фильтрации

Упомянутые фонемы отличаются от всех остальных тем, что после такой фильтрации их участки становятся подобными паузе и содержат большое число точек постоянства. Таким образом, на этих участках разность между числом точек непостоянства и числом точек постоянства будет отрицательной, что позволяет выделить их в массиве таких разностей, построенном для последовательности окон в 256 отсчетов.

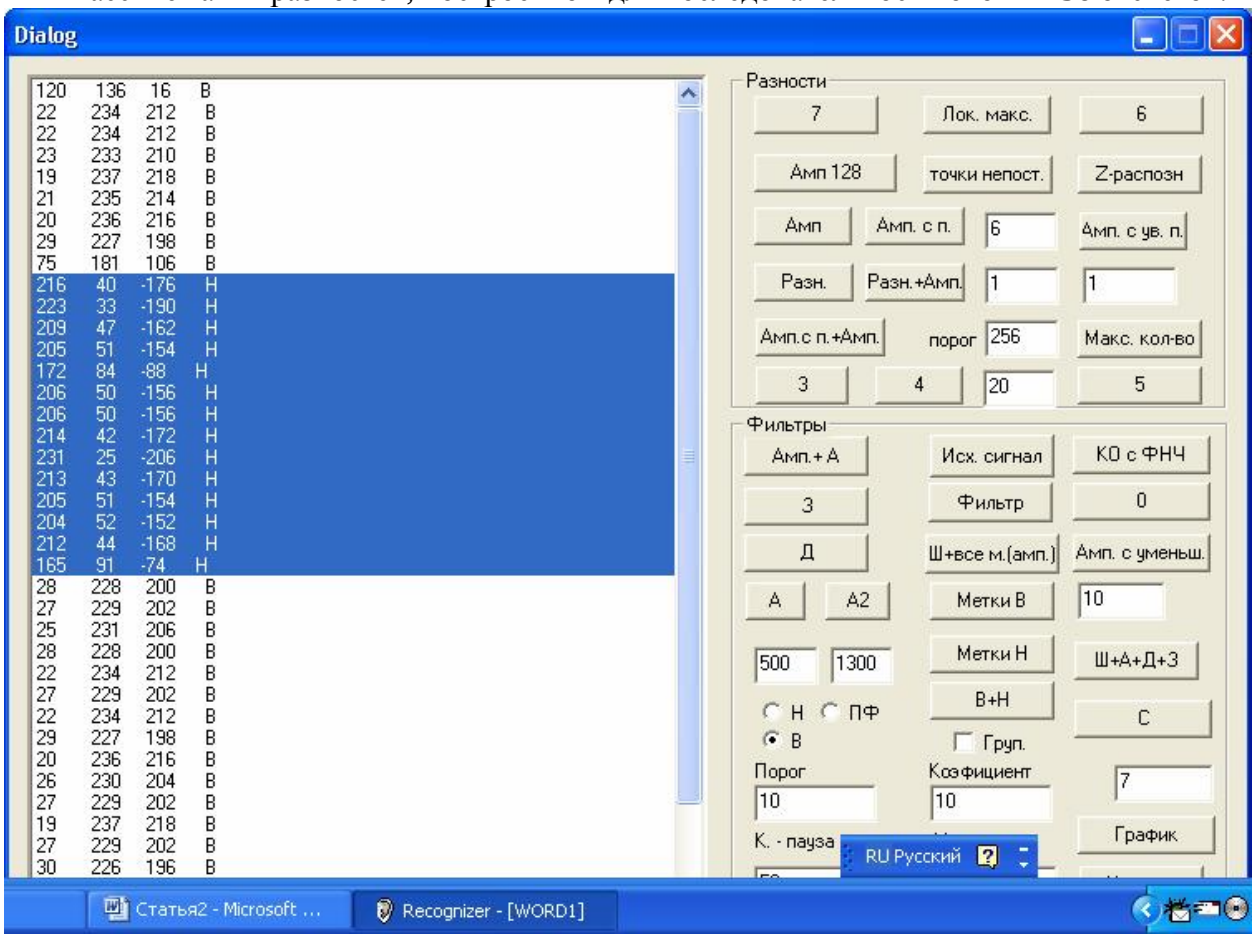

Рисунок 5 – Числовой массив, по которому определяются границы фонемы С

Так же априорно (до общей сегментации) выделяются и распознаются участки, отвечающие дрожащей русской фонеме *Р*.

Способ выделения шипящих и пауз, не использующий фильтрацию

Предлагается m раз последовательно обработать сигнал трехточечным сглаживающим фильтром (6), взяв в качестве m минимальное число, при котором участок шипящей превращается в прямую (для автора этих строк и используемого им микрофона m=25). После этого для записанного речевого сигнала формируется массив величин (4), вычисляемых для последовательности окон по 256 отсчетов. Для этого массива осуществляется «В-Н»-обработка с порогом 0,1 а также обработка «тройками» (7) и «четверками» (8). Начало и конец «Н»-участка отмечаются метками. Они являются концами соответствующей шипящей или паузы.

Далее мы будем использовать величины

V_a - вариация сигнала после a - кратного применения оператора сглаживания (6);

C_a - количество точек постоянства после a - кратного применения оператора сглаживания (6).

Выделение согласных «Ж, З»

Согласные «Ж» и «З» выделяются в речевом сигнале следующим образом. Для того, чтобы перевести искомые участки в число согласных, для которых величина (4) мала, сигнал подвергается 5-кратному сглаживанию. После этого он сегментируется описанным выше способом. Для участков *Ж* и *З* характерна относительно большая вариация и значительное ее уменьшение при 5-кратном сглаживании Поэтому их можно выделить, вычисляя для каждого полученного «Н»-участка величину

$$V^2 / V_5$$

Если она превышает некоторый порог, участок считается отвечающим фонеме *Ж* или *З*, в противном случае – нет.

Описанное сглаживание сигнала перед сегментацией помогает таким образом выделить участки *Ж* и *З*, но приводит к значительному ухудшению результата при разделении *М* и *И*. Поэтому целесообразно проводить окончательную сегментацию для исходного сигнала, сохранив полученную информацию относительно границ участков *Ж* и *З*. Этого мы и достигаем полагая величины (4), отвечающие соответствующим окнам, равными нулю так же как для участков фонем *С,Ш,Щ,К,П,Т,Х,Ф,Ц,Ч, t*.

Детектирование фонемы «Р»

Вот визуализация записи слова «СОРОКА»:

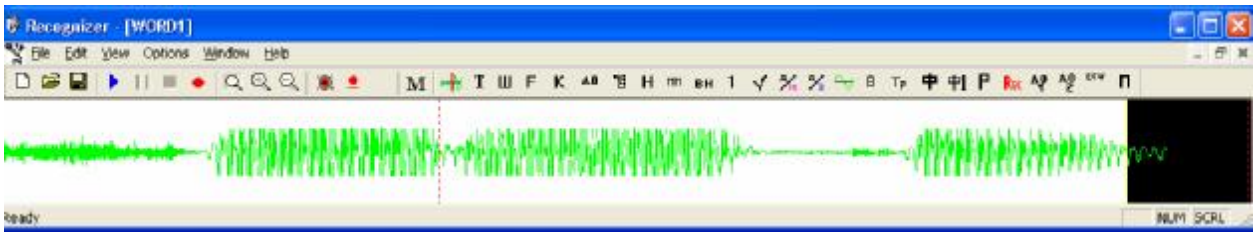

Рисунок 6. Визуализация слова “сорока”

Метка стоит на участке фонемы *Р*. Он характерен достаточно резким падением и последующим возрастанием среднего отклонения. При этом здесь целесообразно использовать более тонкое орудие: среднее отклонение на окне в 128 отсчетов:

$$E = \sqrt{\frac{\sum_{i=1}^{128} |x_i - 128|^2}{128}}$$

С целью детектирования фонемы *Р* строим для всего сигнала массив $E[i]$ таких отклонений, соответствующих последовательным окнам в 128 отсчетов, а затем массив разностей $R[i] = E[i+1] - E[i]$. Далее модифицируем последний массив, полагая $R[i] = 0$ для каждого отрицательного $R[i]$, удовлетворяющего условию $|R[i]| \le p_1$ и каждого положительного $R[i]$, удовлетворяющего условию $R[i] \le p_2$. В получившемся массиве заменяются нулями все отрицательные элементы, за которыми следуют отрицательные (или непосредственно или через любое количество нулевых элементов), и все положительные, за которыми следуют положительные,. Оставшиеся отрицательные $R[i]$ это предполагаемые «входы» в *Р*, оставшиеся положительные $R[i]$ – предполагаемые выходы из *Р*. Фонема считается обнаруженной, и это отмечается проставлением соответствующей метки, если вход и выход достаточно близки, то есть число рассматриваемых 128-окон между ними не превосходит некоторого числа p_3, а также выполняется условие $\min(E_i..E_{i+k}) \le P_4$. Здесь $\min(E_i..E_{i+k})$ - минимальное значение среднего отклонения на участке от входа в «Р» до выхода.

Далее повторяются описанные действия, только в качестве признаков берутся вариация V и отношение V/C. Таким образом, получаются 3 результата по разным критериям. Окончательное решение принимается так: если *P* обнаруживается хотя бы двумя из описанных критериев, и если метки, полученные по этим критериям, находятся достаточно близко друг к другу (на расстоянии не более 768 отсчетов), то считается, что в слове присутствует *P*.

Проблема конца сигнала

При сегментации сигнала на основе выделения высокоамплитудных и низкоамплитудных участков, соответствующих гласным и согласным звукам речи, возникает проблема, связанная с концом сигнала. Любой записанный речевой сигнал в конце затухает постепенно. Поэтому в конце слова, заканчивающегося гласной, машина имеет тенденцию находить низкоамплитудный участок, добавляя при сегментации несуществующую согласную. То же самое происходит при использовании вместо среднего отклонения (5) вариации (4). Вот как, например, выглядит запись слова «МАМА»:

Рисунок 7 – Визуализация слова “мама”

Для устранения этой трудности предлагается следующий метод. Если последний сегмент, полученный при сегментации сигнала оказывается “Н-сегментом”, то для каждого из последовательности 256-окон, считая от начала сегмента, вычисляется величина V_{30}/C_{30}. Если для какого-либо n-го окна она оказывается положительной, а для следующего окна – нулевой, то конечная метка сигнала сдвигается в позицию $beginH + 256n$, где $beginH$ - начало сегмента. Далее на участке от $beginH$ до новой метки вычисляется произведение Vn и сравнивается с некоторым числовым порогом p. Если $Vn < p$ то рассматриваемый «Н-сегмент» удаляется из сегментации, и его начало $beginH$ считается концом записанного сигнала. В противном случае этот сегмент остается и учитывается при распознавании.

Литература

1. Дорохин О.А., Старушко Д.Г., Федоров Е.Е., Шелепов В.Ю. Сегментация речевого сигнала // Искусственный интеллект. – 2000. - №3. – С.450-458.3.
2. Шелепов В.Ю., Ниценко А.В. Амплитудная сегментация речевого сигнала, использующая фильтрацию и известный фонетический состав // Искусственный интеллект. – 2003. - №3. – С.421-426.

3. Ниценко А.В., Шелепов В.Ю. Алгоритмы пофонемного распознавания слов наперед заданного словаря // Искусственный интеллект. – 2004. - №. – С.633-639.
4. В.Ю.Шелепов, А.В. Ниценко. К проблеме пофонемного распознавания // Искусственный интеллект. – 2005. - №4. – С.662-668.

В.Ю. Шелепов, А. В. Ніценко. Структурна класифікація слів російської мови. Нови алгоритми сегментації мовленевого сигналу, розпізнавання фонем та їх класів.
Відомо, що людина може вірно ідентифікувати на послух не більш, ніж 30 % потоку фонем. Але вона має можливість повторить слово, яке чітко вимовлене на невідомої мові (інтерпретуючи звуки в межах звичної фонетичної системи). Це позначає, що пофонемне розпізнавання повіне братися за базу, якщо ми не бажаємо обмежуватися розпізнаванням слів за еталонами, тобто працювати на рівні звукових ієрогліфів.

V.Ju. Shelepov, A.V. Nicenko. Structure classification of Russian words. New algorithms of segmentation, recognitions of phonemes and there classies.
It is known that human can identify correctly no more then 30 % of phonemes stream. But he is able to repeat word which was articulate clear and belongs to unknown language (using familiar phonetic system of cause). It is mean that phoneme recognition must be the base if we do not want to limit ourselves with pattern-recognition, i.e. to work on sound-hieroglyph level.